1991-12

Modeling convergence zone gain on MS-DOS based computers

Jenkins, Alan K.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MODELING CONVERGERGENCE ZONE GAIN ON MS-DOS BASED PERSONAL COMPUTERS

by

Alan K. Jenkins

December 1991

Thesis Advisor: A.B. Coppens
Co Advisor: J.V. Sanders

Approved for public release; distribution is unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
NONE

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Naval Postgraduate School

5. MONITORING ORGANIZATION REPORT NUMBER(S)
N/A

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b. OFFICE SYMBOL (*If applicable*)
Code 3A

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS (*City, State, and ZIP Code*)
Monterey, CA 93943-5000

7b. ADDRESS (*City, State, and ZIP Code*)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
N/A

8b. OFFICE SYMBOL (*If applicable*)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (*City, State, and ZIP Code*)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)
Modeling Convergence Zone Gain on MS-DOS Based Personal Computers

12. PERSONAL AUTHOR(S)
Jenkins, Alan K.

13a. TYPE OF REPORT
Master's Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (*Year, Month, Day*)
December 1991

15. PAGE COUNT
84

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
Convergence Zone, CZ, Convergence Zone Gain, Ray Tracing, Ray Theory, Acoustic Modeling

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

The models for determing convergence zone Gain (G) were developed using a linearized Sound Speed Profile (SSP) and applying ray tracing theory. The SSP was divided into three cases; bilinear, bilinear with isospeed layer, and bilinear with mixed layer. Two analytical solutions were developed using Taylor series and binomial series expansions to determine G, one for the bilinear and bilinear with isospeed layer, and the other for the bilinear with a mixed layer. The solutions for G are exclusively a function of the SSP gradients. Each solution was compared to the solutions from ray tracing and the solutions from the Integrated Carrier ASW Prediction System (ICAPS) (which runs on mainframe computers and requires more data in addition to the SSP). When the SSP's were not too unusual, the solutions for G were fairly close when compared to ray tracing and ICAPS.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
A. B. Coppens

22b. TELEPHONE (*Include Area Code*)
408-646-2941

22c. OFFICE SYMBOL
PH/Cz

Previous editions are obsolete.

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

Modeling Convergence Zone Gain
on MS-DOS Based Computers

by

Alan K. Jenkins
B.S.E.E., Auburn University

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1991

Author: [redacted]
Alan K. Jenkins

Approved by: [redacted]
A.B. Coppens, Thesis Advisor

[redacted]
J.V. Sanders, Co Advisor

[redacted]
J.N. Eagle, Chairman
Antisubmarine Warfare Academic Group

ABSTRACT

The models for determining convergence zone Gain (G) were developed using a linearized Sound Speed Profile (SSP) and applying ray tracing theory. The SSP was divided into three cases; bilinear, bilinear with isospeed layer, and bilinear with mixed layer. Two analytical solutions were developed using Taylor series and binomial series expansions to determine G, one for the bilinear and bilinear with isospeed layer, and the other for the bilinear with a mixed layer. The solutions for G are exclusively a function of the SSP gradients. Each solution was compared to the solutions from ray tracing and the solutions from the Integrated Carrier ASW Prediction System (ICAPS) (which runs on mainframe computers and requires more data in addition to the SSP). When the SSP's were not too unusual, the solutions for G were fairly close when compared to ray tracing and ICAPS.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to thank the personnel in IX03 for their continued support, not only in helping to improve this final product but also in their outstanding teamwork both inside and outside the classroom.

I would also like to thank my advisors for their time spent in helping me to recognize what was too small to include and what was not too small to discard (check the development section). Additionally, they were invaluable in helping me express the results of my research.

I. INTRODUCTION

A. BACKGROUND

Convergence zone (CZ) parameters such as range to CZ, CZ width, and CZ gain are typically obtained using large mainframe computers implementing complex software programs. The U.S. Navy utilizes the Fleet Numerical Oceanographic Center (FNOC), Monterey, California, to predict these parameters. The frequencies of interest, source and receiver depths, Sound Speed Profile (SSP), wave height, time of day, date, and geographic location are used in obtaining the CZ parameters.

In addition to FNOC, the fleet may use the Integrated Carrier ASW Prediction System (ICAPS). ICAPS was designed for use onboard ships. It consists of several databases of environmental data and can provide the user with a cornucopia of acoustic parameters.

Both of these methods require the use of mainframe or "mainframe-like" computers. Several programs for hand-held and desktop calculators have been written in attempts to obtain quick estimates of CZ parameters based on current bathythermograph (BT) data.[1] However, none of the calculator programs is usable on IBM-based disk operating system (MS-DOS) running on personal computers (PC's). With the advent of notebook PC's, and with the palmtop PC on the horizon, a definite need exists for models which will run on an MS-DOS based computer.

[1] See the Bibliography for further information.

B. DESIRABLE QUALITIES FOR COMPUTER MODELS

The following is a list of general guidelines for computer models [Ref. 1:pp.13,14].

- A minimum amount of data should be required to run the program. Additionally, any data used should not be entered more than once.
- Output should give range to the inner edge of the CZ, width of the CZ, and CZ gain. The output should also give a repeat of the inputs.
- The program should run quickly.
- The program should be based on acoustic theory rather than general thumb rules and agree with mainframe prediction models.

C. PURPOSE

The purpose of this study is to provide a MS-DOS based program following the general guideline for desirable qualities for PC models listed above. The model will be based on ray tracing *vice* ray mode theory.

II. CZ RAY TRACING THEORY

A. WHY RAY TRACING?

To simplify the mathematics and programming involved, ray tracing theory was selected over normal mode theory. Ray tracing provides an easy visualization of how sound energy propagates through the ocean, even with complicated SSP and bottom structures. A sound ray can be traced from the source to the desired range without any need for complex calculations. As diffraction of the sound energy becomes more noticeable at lower frequencies, and as ray tracing theory does not consider diffraction, ray tracing becomes more valid as the frequency increases.

Normal mode theory gives a complete description of the sound field. However, it cannot easily handle real boundary conditions. All frequencies can be used but complexity increases with the frequency. [Ref. 2:p.122]

B. CONVERGENCE ZONES

Imagine the sound rays projected from a source at the surface. Some of these rays may get trapped in a surface duct or interact with the bottom. Other rays may be able to penetrate below the SOFAR axis, get refracted upward, and reacquire the surface. Starting from the horizontal and increasing the angle of depression, the first ray which crosses the sound axis and reacquires the surface does so at a range equal to r_{outer}. As the depression angle increases further, the range at which the ray resurfaces first decreases to a minimum, then increases, resweeping back beyond r_{outer}. For an omnidirectional source the reswept area forms an annulus about the source and is acoustically brighter than the surrounding areas. When this brightened area is compared to a reference area at 1 meter from the source, the CZ gain can be calculated.

Figure 1 illustrates the horizontal distance a ray travels in a linear sound speed gradient g.

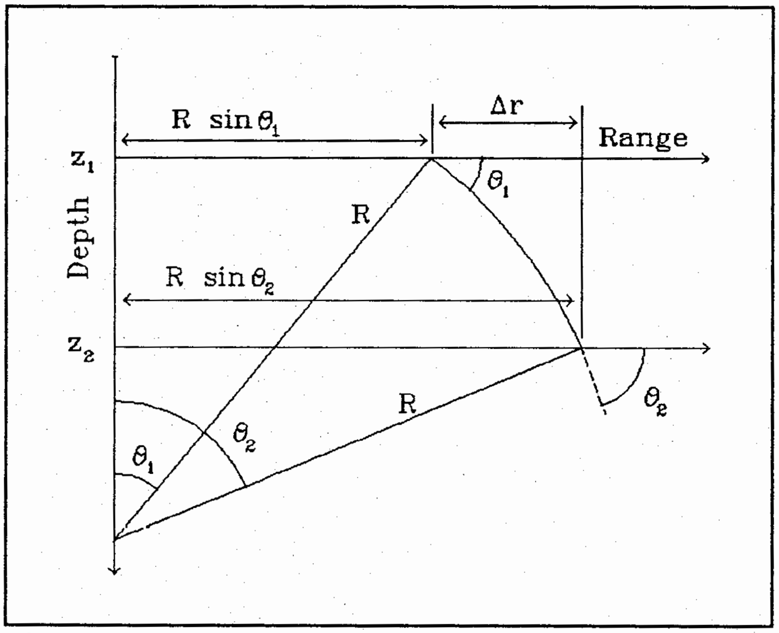


Figure 1 - Horizontal Distances

For the case shown, the sound speed, c, is expressed as a function of depth by $c(z) = c_0(1+gz)$. At depth z_1, the sound speed is c_1, and the downward angle of the ray is θ_1. At depth z_2, the sound speed is c_2, and the downward angle of the ray is θ_2. Snell's law states that $c/(g \cos\theta)$ is constant along the ray path.

The gradient is given by

$$g = \frac{c_1 - c_2}{z_2 - z_1}$$

and radius of curvature R of the ray is

$$R = \frac{c}{g\cos\theta} = \frac{c_1}{g\cos\theta_1} = \frac{c_2}{g\cos\theta_2}$$

The horizontal distance Δr the ray travels and the change in depth Δz are given geometrically by

$$\Delta r = r_2 - r_1 = R(\sin\theta_2 - \sin\theta_1)$$

$$\Delta z = z_2 - z_1 = R(\cos\theta_1 - \cos\theta_2)$$

The ordering of subscripts in the above equations result from the convention of measuring an angle of elevation positive upwards but depth is defined as positive downward.

1. Range to Outer Edge of the Reswept Zone

The ray with initial depression angle of θ_{out} defines the range, r_{outer}, to the outer edge of the reswept zone. At this range, the acoustic energy experiences a sharp drop in intensity. For the bilinear profile illustrated in Fig. 2, $\theta_{out} = 0°$.

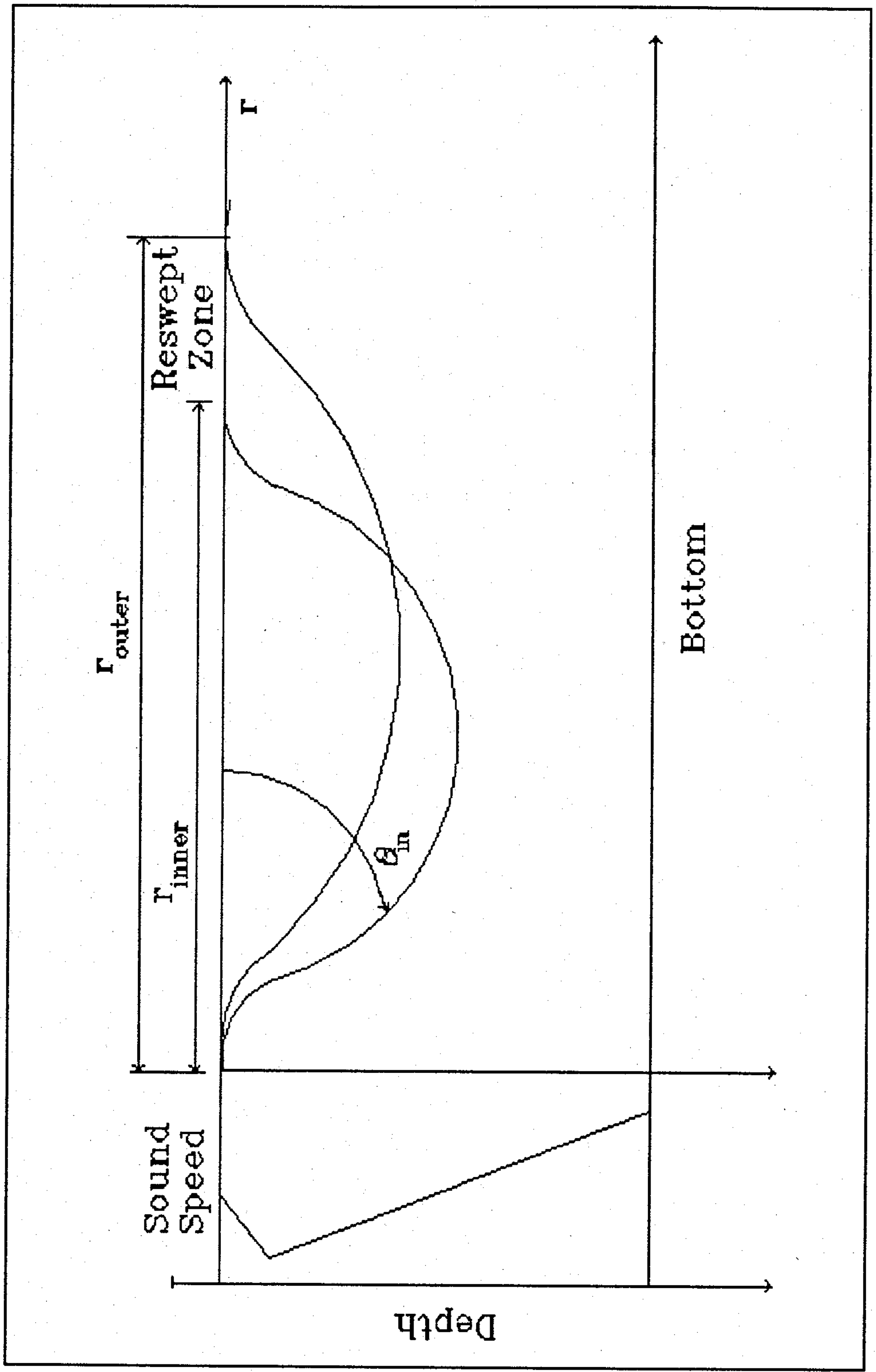


Figure 2 - Bilinear profile sound rays of interest

2. Range to Inner Edge of the Reswept Zone

The ray with a depression angle of θ_{in} defines the range, r_{inner} to the inner edge of the reswept zone. Because of the strong convergence of neighboring rays, with the formation of a caustic (where sound rays cross each other) at the inner edge of the CZ, there is a strong enhancement of the intensity. Additionally, ray theory does not provide a suitable description of this part of the problem.

3. Range to the CZ

The range to the CZ zone will be defined as the range r_{outer} [Ref. 1 :p.51]. For the remainder of the discussion, the reswept zone will be considered the CZ, and bottom interactions will not be considered. This follows a consistent course as developed in earlier models which have also ignored bottom interactions [Ref.1:p.51].

4. Width of the Convergence Zone

The CZ width, Δr_{CZ} , is the distance from the outer edge of the reswept zone to the inner edge of the reswept zone. The width of the CZ is thus taken to be

$$\Delta r_{CZ} = r_{outer} - r_{inner} \tag{1}$$

5. Convergence Zone Gain

The general definition for Transmission Loss (TL) is

$$TL = 10\log\left(\frac{I(1)}{I(r)}\right)$$

where I(1) is the intensity of the sound beam at r = 1 meter, and I(r) is the intensity of the sound beam at r meters from the source.

By conservation of energy, the following relationships for the intensities and areas through which the rays pass normally can be made.

$$I(1)A(1) = I(r)A(r)$$

where A(1) is the cross-sectional surface area for the sound beam as illustrated in Fig. 3, and A(r) is the cross-sectional area for the beam at r. The geometric TL is now

$$TL_g = 10\log\left(\frac{A(r)}{A(1)}\right)$$

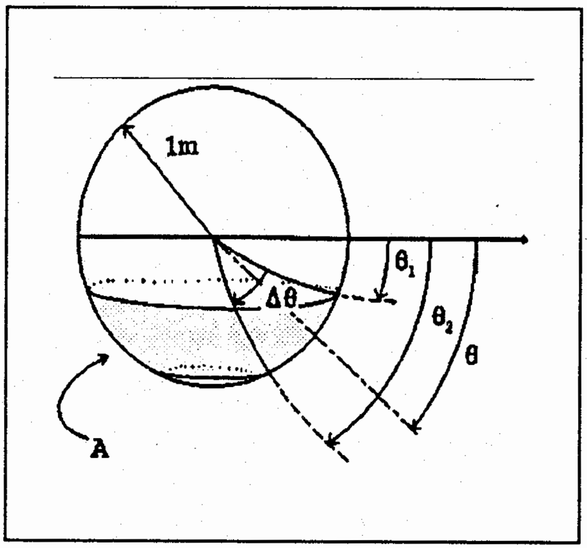


Figure 3 - Reference Area Diagram for CZ Gain

The area A(1) is

$$A(1) = 2\pi\,\Delta\theta\,\cos\theta_{in}$$

Figure 4 pictures a cross sectional view of the area contained by the CZ. The intensity will be assumed to be uniform over this area. The area normal to the arriving rays of the CZ annulus [Ref. 1:p.62] is

$$A(r_{inner}) = 2\pi\, r_{inner}\, h$$

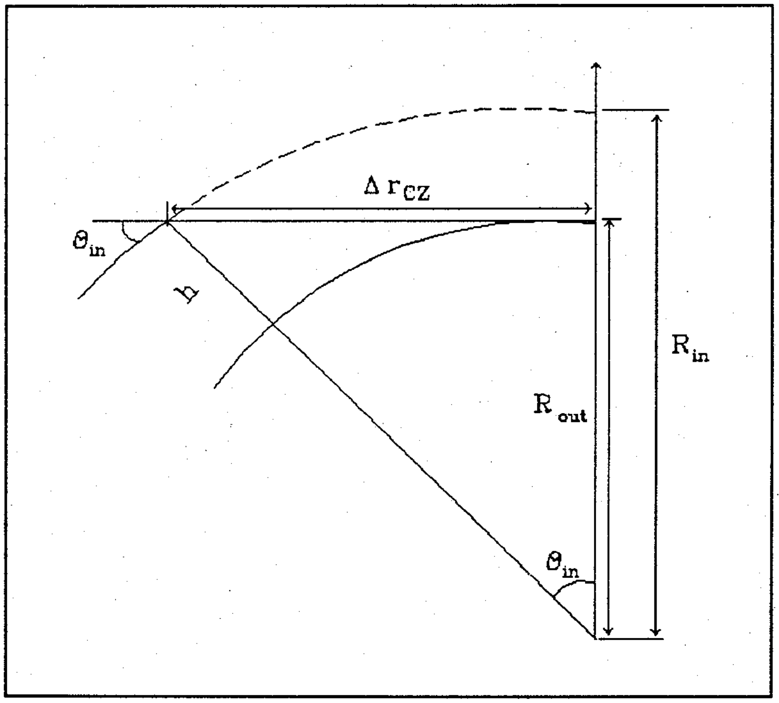


Figure 4 - CZ reference area (cross section)

TL_g becomes,

$$TL_g = \frac{2\pi\,\Delta\theta\cos\theta_{in}}{2\pi r_{inner} h}$$

$$\frac{\Delta r_{CZ}}{R_{out}} = \tan\theta_{in}$$

$$\frac{\Delta r_{CZ}}{R_{in}} = \sin\theta_{in}$$

$$h = \frac{\Delta r_{CZ}}{\sin\theta_{in}}\left(1 - \cos\theta_{in}\right)$$

When the small angle approximation of $\Delta\theta = \theta_{in} << 1$ is substituted, and

$$A(1) \doteq 2\pi\theta_{in}$$

$$A(r_{inner}) \doteq \pi r_{inner} \Delta r_{CZ} \theta_{in}$$

TL_g is now

$$TL_g = 20\log r_{inner} + 10\log\left(\frac{2}{\Delta r_{CZ} \cdot r_{inner}}\right)$$

CZ Gain (G) is the difference between the TL_g from spherical spreading [20log(r)] and what is actually observed for the geometrical spreading loss. (There are two definitions for G. The first definition includes additional losses such as absorption in G, and the second definition considers absorptive losses separately [Ref. 3:p.408] [Ref. 4:p.7.1]. As a frequency independent model will be developed, the second definition for G will be used.)

So G is now

$$G = 10\log\frac{2r_{inner}}{\Delta r_{CZ}} \tag{2}$$

In real ocean conditions, the intensity within the CZ has an appearance similar to a Rayleigh distribution. The intensity rises sharply from zero at r_{inner} and gradually declines to zero at r_{outer} . Because the exact distribution is unknown and a single number is desired for the CZ gain, G will be assumed to be uniformly distributed over Δr_{CZ} , and (2) will give the average value of G. Simplifying G to an average value follows earlier work done in similar models [Ref.1:p.40].

III. MODEL DEVELOPMENT

A. GENERAL

Several approximations are used throughout the development to reduce equations to a manageable and easily understandable form. The approximations used are contained in Appendix A.

The convention used when labeling angles, sound speeds, and radii of curvature is the subscript is always tied to the gradient. As illustrated in Fig. 5, for the deep gradient g_1, the ray entering this region at a depth z_1 has a radius of curvature R_1 and an initial angle entering this region of θ_1.

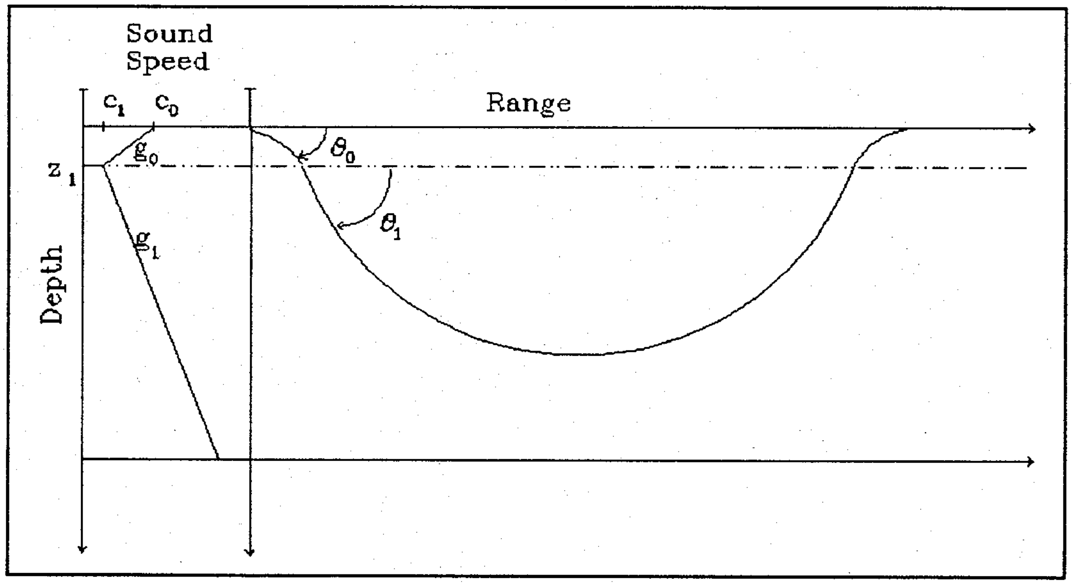


Figure 5 - Reference diagram for subscript conventions.

The three SSP's which will be examined are bilinear, bilinear with isospeed layer, and bilinear with mixed layer. Figure 6 illustrates the three profile types.

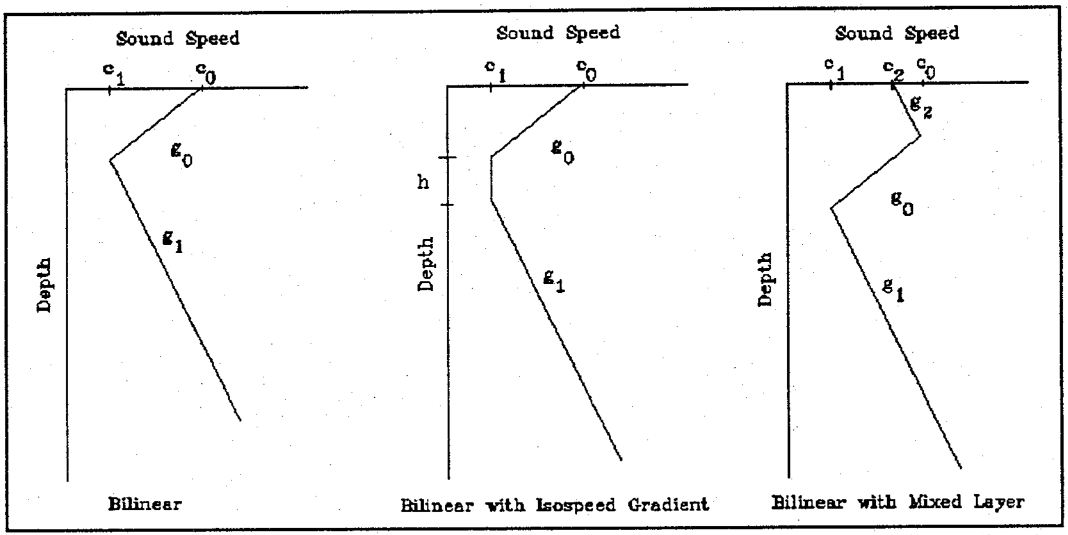


Figure 6 - Bilinear, Bilinear with Isospeed Layer, and Bilinear with Mixed Layer SSP's

B. BILINEAR SSP

1. Range to the Outer Edge of the CZ

The ray reacquiring the surface at r_{outer} has an initial depression angle $\theta_{out} = 0°$.

2. Range to the Inner Edge of the CZ

Examination of Fig. 5 shows that the range from the source (at the surface) to where a ray with a depression angle θ_0 reacquires the surface can be expressed as:

$$r = 2\left[R_0(\sin\theta_1 - \sin\theta_0) + R_1 \sin\theta_1\right] \tag{3}$$

Using Snell's Law:

$$\frac{c_0}{\cos\theta_0} = \frac{c_1}{\cos\theta_1}$$

The radius of curvature for the lower layer R_1 can be expressed in terms of the radius of curvature for the upper layer R_0,

$$R_1 = \frac{g_0}{g_1} R_0$$

and when R_1 is eliminated from the range equation (3),

$$r = 2R_0\left[\left(1 + \frac{g_0}{g_1}\right)\sin\theta_1 - \sin\theta_0\right]$$

Again using Snell's Law, a relationship can be found for the θ_0 and θ_1 ,

$$\sin\theta_1 = \sqrt{1 - \cos^2\theta_1}$$

$$\sin\theta_1 = \sqrt{1 - \frac{c_1^2}{c_0^2}\cos^2\theta_0}$$

Simplifying the range equation to

$$r = 2R_0\left[\left(1+\frac{g_0}{g_1}\right)\sqrt{1 - \frac{c_1^2}{c_0^2}\cos^2\theta_0} - \sin\theta_0\right]$$

$$r = 2\frac{c_0}{g_0}\left[\left(1+\frac{g_0}{g_1}\right)\sqrt{\frac{1}{\cos^2\theta_0} - \frac{c_1^2}{c_0^2}} - \tan\theta_0\right] \tag{4}$$

Manipulation of (the above equation) to obtain r_{inner} is cumbersome but direct. Differentiating r with respect to θ_0 and imposing

$$\left.\frac{dr}{d\theta_0}\right|_{\theta_0=\theta_{in}} = 0$$

enables θ_{in} to be solved analytically and directly input into (3) to solve for r_{inner} .

Without further simplification the solution for θ_0 yields a difficult transcendental equation for θ_{in} . However, because θ_0 is generally significantly less than 20° (0.3491 radian) the trigonometric functions can be replaced by Taylor series expansions and terms of third and higher order can be discarded. By replacing the trigonometric functions, (4) simplifies to

$$r = 2\,\frac{c_0}{g_0}\left[\left(1+\frac{g_0}{g_1}\right)\sqrt{\frac{1}{\left(1-\frac{1}{2}\theta_0^2\right)^2}-\frac{c_1^2}{c_0^2}}-\theta_0\right]$$

Further manipulation of the range equation using the binomial expansions in Appendix A yields

$$r = 2\,\frac{c_0}{g_0}\left[\left(1+\frac{g_0}{g_1}\right)\sqrt{1+\theta_0^2-\frac{c_1^2}{c_0^2}}-\theta_0\right] \tag{5}$$

Convenient parameters can be defined:

$$\gamma \equiv \frac{c_1}{c_0} \tag{6}$$

$$\alpha^2 \equiv 1-\gamma^2 \tag{7}$$

and,

$$\Gamma \equiv 1 + \frac{g_0}{g_1} \tag{8}$$

The final form of the range equation

$$r = 2\frac{c_0}{g_0}\left[\Gamma\sqrt{1-\gamma^2+\theta_0^2} - \theta_0\right] \tag{9}$$

Now, differentiation yields

$$\frac{dr}{d\theta_0} = 2\frac{c_0}{g_0}\left[\frac{\theta_0\Gamma}{\sqrt{1-\gamma^2+\theta_0^2}} - 1\right]$$

Note that for $\theta_0 = 0$, $dr/d\theta_0$ is negative, so the range is decreasing with an increasing θ_0. The minimum range, r_{inner} is found by determining the value of $\theta_0 = \theta_{in}$ for which $dr/d\theta_0$ vanishes.

$$\left.\frac{dr}{d\theta_0}\right|_{\theta_0=\theta_{in}} = 2\frac{c_0}{g_0}\left[\frac{\theta_{in}\Gamma}{\sqrt{1-\gamma^2+\theta_{in}^2}} - 1\right] = 0$$

Manipulating this equation gives

$$1 = \frac{\theta_{in}\Gamma}{\sqrt{1-\gamma^2+\theta_{in}^2}}$$

$$\theta_{in}^2\left(\Gamma^2-1\right)-\left(\gamma^2-1\right)=0$$

and finally,

$$\theta_{in} = \sqrt{\frac{1-\gamma^2}{\Gamma^2-1}} \qquad (10)$$

Substituting into (9) yields an r_{inner} of

$$r_{inner} = 2\frac{c_0}{g_0}\left[\Gamma\sqrt{\alpha^2+\frac{\alpha^2}{\Gamma^2-1}}-\sqrt{\frac{\alpha^2}{\Gamma^2-1}}\right]$$

Reducing we have

$$r_{inner} = 2\frac{c_0}{g_0}\sqrt{\frac{\alpha^2}{\Gamma^2-1}}\left[\Gamma^2-1\right]$$

and finally,

$$r_{inner} = 2\frac{c_0}{g_0}\sqrt{\alpha^2\left(\Gamma^2 - 1\right)} \tag{11}$$

3. CZ Gain

With substitution of r_{inner} and r_{outer} into (1), the CZ width can be expressed as

$$\Delta r_{CZ} = 2\frac{c_0}{g_0}\left[\Gamma\sqrt{\alpha^2} - \sqrt{\alpha^2\left(\Gamma^2 - 1\right)}\right]$$

$$\Delta r_{CZ} = 2\frac{c_0}{g_0}\sqrt{\alpha^2}\left[\Gamma - \sqrt{\Gamma^2 - 1}\right] \tag{12}$$

Substitution into the CZ gain, equation (2), gives

$$G = 10\log\left(\frac{2\sqrt{\Gamma^2 - 1}}{\Gamma - \sqrt{\Gamma^2 - 1}}\right) \tag{13}$$

Observe that G is only a function of the upper and lower gradients and the lack of dependance upon sound speed, depth (although these are implicitly contained in the gradients), and the ray departure angles.

C. BILINEAR WITH ISOSPEED LAYER SSP

1. Range to the Outer Edge of the CZ

As with the previous SSP, r_{outer} is determined by using an initial depression angle of $\theta_0 = 0°$.

2. Range to the Inner Edge of the CZ

Adding an isospeed layer of thickness h to the bilinear SSP gives a new range equation

$$r = 2\left[R_0(\sin\theta_1 - \sin\theta_0) + h\cot\theta_1 + R_1\sin\theta_1\right] \qquad (14)$$

Manipulation of the R's and trigonometric functions yields

$$\frac{r}{2} = \frac{c_0}{g_0}\left[\Gamma\sqrt{\frac{1}{\cos^2\theta_0} - \gamma^2} - \tan\theta_0\right] + \frac{\gamma h}{\sqrt{\frac{1}{\cos^2\theta_0} - \gamma^2}}$$

Making use of series expansions from Appendix A,

$$\frac{r}{2} = \frac{c_0}{g_0}\left[\Gamma\sqrt{1 - \gamma^2 + \theta_0^2} - \theta_0\right] + \frac{\gamma h}{\sqrt{1 - \gamma^2 + \theta_0^2}} \qquad (15)$$

An additional definition which will prove to be useful is,

$$\epsilon = \frac{g_0 \gamma h}{c_0} \tag{16}$$

To get r_{inner},

$$\left.\frac{dr}{d\theta_0}\right|_{\theta_0 = \theta_{in}} = 0$$

Which gives the following equation,

$$0 = \frac{\Gamma \theta_{in}}{\sqrt{\alpha^2 - \theta_{in}^2}} - 1 - \frac{\epsilon \theta_{in}}{\left(\alpha^2 - \theta_{in}^2\right)^{\frac{3}{2}}}$$

and by factoring out the radicals,

$$\alpha^2 + \theta_{in}^2 = \Gamma^2 \theta_{in}^2 - \frac{2 \epsilon \Gamma \theta_{in}^2}{\alpha^2 + \theta_{in}^2} + \frac{\epsilon^2 \theta_{in}^2}{\left(\alpha^2 + \theta_{in}^2\right)^2}$$

and finally,

$$0 = \theta_{in}^6 \left[\Gamma^2 - 1\right] + \theta_{in}^4 \alpha^2 \left[2\Gamma^2 - 3 + 2\epsilon\Gamma\right] + \theta_{in}^2 \alpha^4 \left[\Gamma^2 - 3 + \frac{2\epsilon\Gamma}{\alpha^2} + \frac{\epsilon^2}{\alpha^4}\right] - \alpha^6 \tag{17}$$

Note that this equation is cubic in θ_{in}^2 whereas the bilinear case the corresponding equation was linear in θ_{in}. Additionally, as h approaches zero, this reduces to equation (10). To reduce this cubic equation to a more manageable form, the relative size of the coefficients for θ_{in} will be considered. Using historical data on SSP's from the Atlantic, Pacific, and Mediterranean the following can be related [Ref. 1:pp.30,67]:

$$\gamma < 1$$

$$\alpha \ll 1$$

$$g_0 \ll 1$$

$$0 < h < 250$$

$$c_0 \approx 1500$$

Thus, $\epsilon \ll 1$

Because ϵ is much less than 1, the ϵ^2 term will be discarded. By discarding this term, an equation which is quadratic with respect to θ_{in} results and can be expressed:

$$0 = \theta_{in}^4 \left[\Gamma^2 - 1\right] + \theta_{in}^2 \alpha^2 \left[\Gamma^2 - 2 - \frac{2\Gamma\epsilon}{\alpha^2}\right] - \alpha^4 \qquad (18)$$

And finally,

$$\theta_{in} = \sqrt{\frac{\alpha^2}{2(\Gamma^2-1)}}\left[-\left(\Gamma^2-2-\frac{2\epsilon\Gamma}{\alpha^2}\right)+\sqrt{\left(\Gamma^2-2-\frac{2\epsilon\Gamma}{\alpha^2}\right)^2+4(\Gamma^2-1)}\right]^{\frac{1}{2}} \qquad (19)$$

3. Width of CZ and CZ Gain

As expected, when ϵ (which is directly proportional to h) approaches zero this equation reduces to (10). Equation (19) does give the analytical solution to the isospeed case but is overly complicated when ϵ (thus h) is small (less than 250 m). Equation (10) will be used in the analysis as the solution for Δr_{CZ} and G.

D. BILINEAR WITH MIXED LAYER SSP

1. Range to Outer Edge of the CZ

With the addition of a mixed layer to the bilinear SSP, and remembering $\theta_0 = 0°$, the initial depressions angle θ_2 for a source at the surface is expressed by,

$$\theta_2 = \operatorname{acos}\left(\frac{c_2}{c_0}\right)$$

2. Range to the Inner Edge of the CZ

Figure 6 illustrates the SSP with the addition of a mixed layer. The range equation is now

$$r = 2R_0\left[\left(1 + \frac{g_0}{g_1}\right)\sin\theta_1 - \sin\theta_0 + \frac{g_0}{g_2}(\sin\theta_2 - \sin\theta_0)\right] \tag{20}$$

Using the following definitions

$$H = 1 + \frac{g_0}{g_2} \tag{21}$$

$$\eta = \frac{c_2}{c_0} \tag{22}$$

$$\beta^2 = 1 - \eta^2 \tag{23}$$

results in the range equation becoming

$$r = 2R_0\left[\Gamma\sin\theta_1 - H\sin\theta_0 + \frac{g_0}{g_2}\sin\theta_2\right] \tag{24}$$

$$r = 2\frac{c_0}{g_0}\left[\Gamma\sqrt{1 - \gamma^2 + \theta_0^2} - H\theta_0 + \frac{g_0}{g_2}\sqrt{1 - \eta^2 + \theta_0^2}\right] \tag{25}$$

(25) should reduce to (9) when c_2 is equal to c_0, and g_2 is equal to g_1. Using these conditions, (25) does reduce to (9) and consequently can be solved by use of (10).

Again, finding the angle, $\theta_0 = \theta_{in}$ at which $r = r_{inner}$,

$$0 = \frac{\Gamma\theta_{in}}{\sqrt{1-\gamma^2+\theta_{in}^2}} - H + \frac{g_0}{g_2}\frac{\theta_{in}}{\sqrt{1-\eta^2+\theta_{in}^2}}$$

so that

$$\theta_{in} = H\left[\frac{\Gamma}{\sqrt{1-\gamma^2+\theta_{in}^2}} + \frac{g_0}{g_2}\frac{1}{\sqrt{1-\eta^2+\theta_{in}^2}}\right]^{-1}$$

The solution for θ_{in} in this form is very formidable. However, if the surface sound speed is approximately equal to the sound axis speed, then $\alpha^2 \approx \beta^2$. θ_{in} can be expressed as

$$\theta_{in} = H\left[\frac{\Gamma}{\sqrt{\alpha^2+\theta_{in}^2}} + \frac{g_0}{g_2}\frac{1}{\sqrt{\alpha^2+\theta_{in}^2}}\right]^{-1}$$

$$\theta_{in} = H\alpha^2\left[\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2\right]^{-\frac{1}{2}} \tag{26}$$

r_{inner} can be expressed by:

$$r_{inner} = 2\frac{c_0}{g_0}\left[\Gamma\sqrt{\alpha^2 + \frac{H^2\alpha^2}{\left(\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2\right)}} - \frac{H^2\alpha}{\sqrt{\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2}} + \frac{g_0}{g_2}\sqrt{\alpha^2 + \frac{H^2\alpha^2}{\left(\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2\right)}}\right]$$

Which reduces to,

$$r_{inner} = 2\frac{c_0}{g_0}\sqrt{\left(1-\gamma^2\right)\left[\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2\right]} \tag{27}$$

Additionally, r_{outer} becomes:

$$r_{outer} = 2\frac{c_0}{g_0}\sqrt{1-\gamma^2}\left[\Gamma + \frac{g_0}{g_2}\right] \tag{28}$$

3. CZ Gain

The width of the CZ is now:

$$\Delta r_{CZ} = 2\frac{c_0}{g_0}\sqrt{1-\gamma^2}\left[\left(\Gamma + \frac{g_0}{g_2}\right) - \sqrt{\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2}\right] \tag{29}$$

Substitution of r_{inner}, and Δr_{CZ} into (2) yields,

$$G = 10\log\left[\frac{2\sqrt{\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2}}{\left(\Gamma + \frac{g_0}{g_2}\right) - \sqrt{\left(\Gamma + \frac{g_0}{g_2}\right)^2 - H^2}}\right] \tag{30}$$

Note the similarity between (13) and (30) in that the only dependance is upon the gradients.

E. COMPUTER MODELS

In developing the computer programs to validate θ_{in} and G, the bilinear SSP was the basic case of interest with the mixed layer or isospeed layer being extensions. For this reason several conditions are placed upon the computer model inputs.

- c_2 is always less than c_0.

- c_1 is always less than c_0.
- h is never negative.
- The mixed layer is at least 1 m.
- The bottom gradient is 0.017 m/s/m.

Appendices B, C, and D contain the Fortran listings for the computer programs used to validate the equations for θ_{in} and G for the bilinear, bilinear with isospeed layer, and bilinear with mixed layer SSP's respectively. Combinations of sound speed, gradients, and thickness of isospeed layer were used where appropriate. But in order to evaluate the different cases, a range for the input values needed to be determined. Theses ranges were obtained from historical SSP's produced by ICAPS and contained in [Ref.1]. Profiles are from the Pacific (40N 140W), Atlantic (31N 69W), and Mediterranean (36N 18E) oceans for the months of February, May, August, and November [Ref. 1:pp.16-30]. Each estimate was evaluated using the following ranges of input where applicable:

- c_0 ranges from 1489 to 1541
- c_1 ranges from 1475 to 1525
- c_2 ranges from 1485 to 1525
- g_0 ranges from 0.01 to 0.246
- g_1 ranges from 0.003 to 0.017
- g_2 ranges from 0.01 to 0.06
- h ranges from 0 to 250

These sound speed, gradient, and h ranges were programmed into the validation programs, and equations (10)/(13) or (26)/(30), where appropriate, were evaluated against the solution obtained by ray tracing to determine θ_{in} , r_{inner}, Δr_{CZ} , and G. The input data range increment size varied depending on available storage space for the output files. As the number of variables increased the increment size was adjusted to give the smallest possible value while maintaining adequate space for file storage. A percent error for G was calculated using the following algorithm:

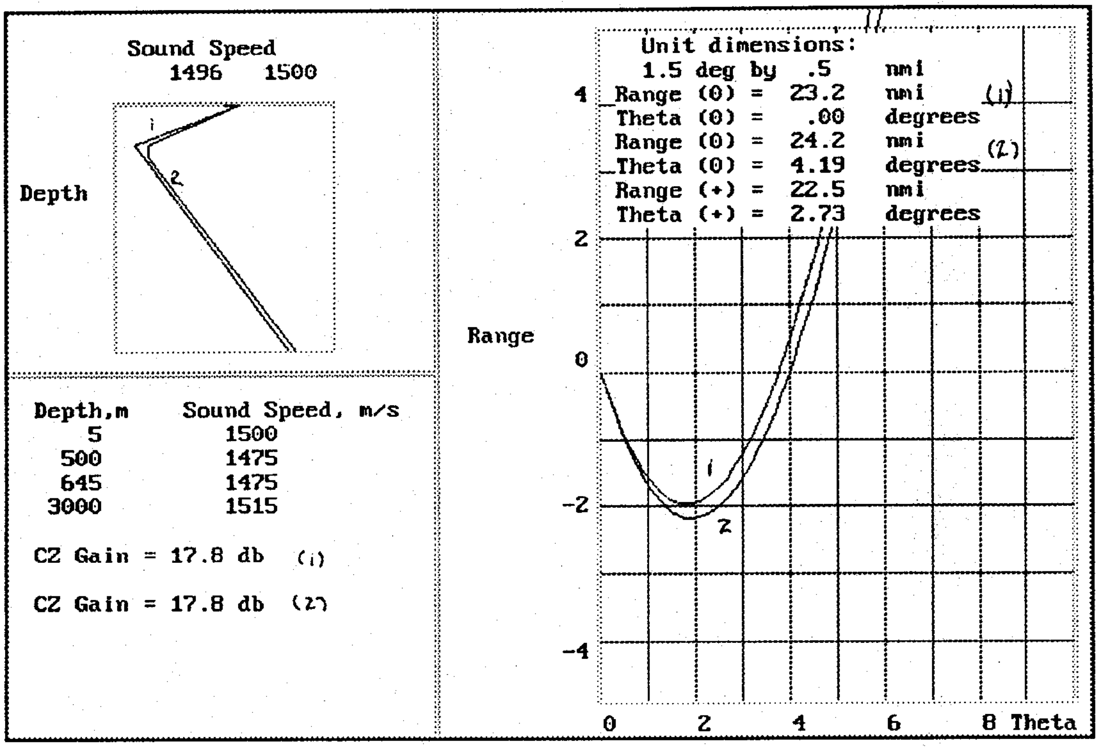
$$\%\ \text{error} = 100 \times \frac{\text{estimate} - \text{ray tracing}}{\text{ray tracing}}$$

The historical data were input into ICAPS in calculating an average CZ gain. [Ref.1:pp.75] Appendix E contains the Fortran listing for the computer program, CZGAIN, used to compare the historical data from ICAPS to the results for G.

Figure 7 depicts a typical display for CZGAIN. The user inputs c_2, c_0, d_0, c_1, d_1, h, and bottom depth. Once the user inputs the initial values, he may then manipulate those values with keyboard entries. Appendix F contains a user's guide for operation of CZGAIN. All of the data are color coded to the respective SSP (which labeled 1 and 2 for the black and white depiction). The unit dimensions for the Range vs. Theta are defined at the top, and the graph is scaled so that each curve starts at Theta(0), Range(0). As Theta (the depression angle at the source) is swept from Theta(0) through Theta(10) (0° to 15° as shown in Fig. 7) the range at which that ray resurfaces is plotted. Observe in Fig. 7 that the unit dimensions remain the same for both curves, and each curve starts from the same point on the graph. The starting reference points (Range(0) and Theta(0)) are also color coded to their respective SSP's. Finally, the G's (also color coded) are presented. The bilinear approximation for r_{inner} and θ_{in} are given by Range(+) and Theta(+). Notice that as the SSP's (1) and (2) become more similar the Range vs. Theta curves approach each other, and the minimal effect from the isospeed thickness upon the Range vs. Theta curves.

The programs contain in Appendix B and C contain the basic algorithms and equations used for development of the analytical part of CZGAIN. The coding for the equations may more easily be extracted from any of these programs.

All of the programs were written using Microsoft's Fortran Version 5.0 on a Compaq Deskpro 386/33 Model 84 computer. The programs contained in Appendices B, C, and D have also run on the Naval Postgraduate School's mainframe after a few small modifications were made (e.g. the output files names were modified). A special software patch was obtained from Microsoft to allow the user to manipulate the data parameters without having to use the "Enter" key.



Depth,m	Sound Speed, m/s
5	1500
500	1475
645	1475
3000	1515

CZ Gain = 17.8 db (1)

CZ Gain = 17.8 db (2)

```
Unit dimensions:
 1.5 deg by  .5     nmi
Range (0) =  23.2   nmi      (1)
Theta (0) =   .00   degrees
Range (0) =  24.2   nmi      (2)
Theta (0) =  4.19   degrees
Range (+) =  22.5   nmi
Theta (+) =  2.73   degrees
```

Figure 7 - CZGAIN Display Depiction

IV. RESULTS AND VALIDATION

A. GENERAL

While specific results will be presented later, overall the estimates for the CZ gain closely approximated the values obtained by ray tracing and from ICAPS.

If a difference of more than 3 dB existed between the estimate and the solution from ray tracing or ICAPS, the estimate was considered invalid. The 3 dB criterion was chosen because the numerous estimations to real ocean conditions made not only by the models developed here but also by ray tracing. A tighter requirement upon the models serves no purpose other than to require a higher degree of accuracy from the estimate, forcing additional complexity on the estimate.

The greatest potential for range errors (and thus errors in the gain calculations) occurs when the gradients are very small [Ref.1:p.66]. When the gradients are small an extremely high degree of accuracy is required for θ_0 to determine θ_{in} . The accuracy is required to diminish the effects of roundoff error and decrease the incrementation size for θ_0 . As c_1 and/or c_2 approaches c_0 the gradients approach zero (assuming that the respective sound speed depths remain constant). When c_0 and c_1 and/or c_0 and c_2 were less than about 2 m/s apart the model becomes invalid. Modifying the software to obtain the needed degree of accuracy for θ_0 was not considered to be necessary as this closeness of sound speeds is not a common occurrence. During validation, the sound speeds were not permitted to be less than 2 m/s apart.

While the closeness of the sound speeds rarely occurs in the ocean, it does demonstrate a weakness in the model which would not have been discovered without model validation.

In comparing this model to the results from earlier models [Ref.1:p.75] it should be noted that earlier results used a source and receiver at 18.3 m. Also ICAPS considers reflection from the surface, absorption, bottom interactions, and coherent ray interaction.

The tables to be presented contain some representative samples of results for

- G from (13) or (30)
- Ray Tracing (RT)
- Previous calculator models (CALC) [Ref.1:p.75]
- Averaged ICAPS values [Ref.1:p.75]

Not all of the data are presented (the entire data set from the validation exceeds 2000 pages). The data presented are a subset from the complete date set which most closely matches the profiles used by earlier models. The tables are intended to show results from earlier estimations (CALC) as well as results from more complete models (ICAPS).

An exact fit the SSP's used by CALC and ICAPS could not be made as they are linearized by different methods. In determining G from equations (13) and (30) the best possible fit to the SSP's was made.

The SSP's from [Ref.1] were not included because only general types of SSP and not specific cases of profiles were not examined. The SSP's used are readily available in [Ref.1].

The ICAPS data is an averaged value and is obtained from [Ref.1] by the following process:

> Two levels were chosen ... which bracket the majority of the TL points within the annulus. The approximate midpoint between those levels was then picked as "the" TL for that CZ. [Ref.1:p.40]

B. BILINEAR SSP

Table 1 presents the data resulting from use of equation (13) along with the RT, CALC, and ICAPS data. The SSP's numbered 18 through 20 [Ref.1:pp.18-20] correspond to SSP's from the Pacific Ocean for May, August, and November respectively. Note the agreement between the results from equation (13) and RT. Also note that overall a closer estimate to the averaged ICAPS data is obtained with equation (13) than was obtained with CALC.

TABLE I - BILINEAR SSP CZ GAINS IN DB (PACIFIC OCEAN)

SSP Number	(13)	RT	ICAPS	CALC
18	17.2	17.2	16.4	15.8
19	18.2	18.2	15.8	18.0
20	15.3	15.3	14.0	10.0

Table II presents the data from use of equation (13) along with the RT, CALC, and ICAPS data. For the SSP's numbered 26 through 28 [Ref.1:pp.26-28] corresponding to the SSP's from the Mediterranean Sea for May, August, and November respectively. The February SSP from the Mediterranean Sea did not have the proper gradient structure to allow the formation of a convergence zone.

TABLE II - BILINEAR SSP CZ GAINS IN DB (MEDITERRANEAN SEA)

SSP Number	(13)	RT	ICAPS	CALC
26	17.5	17.7	14.0	18.0
27	19.6	19.7	17.7	17.0
28	15.4	15.4	13.0	17.0

As with the Pacific Ocean data, the results from equation (13) agree closely with ray tracing (less than 1% on average). Table III presents the percent error between the averaged ICAPS data and equation (13), ray tracing, and CALC.

TABLE III - BILINEAR SSP - AVERAGE PERCENT ERRORS FROM ICAPS

(13)	RT	CALC
13.9	14.3	18.2

C. BILINEAR WITH ISOSPEED LAYER SSP

The approximation of the bilinear case (13) was used rather than (19) because of the complexity involved in evaluating θ_{in} by (19). Although (19) could be programmed on a computer as (13) and (30) were, the thickness of the isospeed layer does not enter significantly into the range equations (unless h is greater than 250 m.). The appropriate SSP's [Ref.1:pp.21-24] are labeled 21 through 24 in Table IV. These SSP's correspond to profiles from the Atlantic Ocean for February, May, August, and November.

TABLE IV - BILINEAR WITH ISOSPEED LAYER SSP'S
(ATLANTIC OCEAN)

SSP Number	(13)	RT	ICAPS	CALC
21	16.7	16.7	15.5	5.0
22	15.8	15.7	16.5	16.0
23	17.4	17.4	16.0	18.0
24	15.2	15.2	12.0	17.0

Again, the solution obtained from use of (13) agrees closely with the solution obtained from ray tracing. The Atlantic Ocean SSP's were more difficult to segment because of a more complex sound speed structure. This difficulty has also been noted in previous model development [Ref.1:p.66]. Table V presents the percent error between the averaged ICAPS data and equation (13), ray tracing, and CALC.

TABLE V - BILINEAR WITH ISOSPEED LAYER SSP - AVERAGE PERCENT ERRORS FROM ICAPS

(13)	RT	CALC
11.9	11.4	31.2

D. BILINEAR WITH A MIXED LAYER SSP

Table VI presents the data resulting from use of equation (30) along with the ray tracing, ICAPS, and CALC results. Only one SSP, numbered 17 [Ref.1:p.17], was appropriate for the bilinear with mixed layer SSP case. This SSP corresponds to the SSP from the Pacific Ocean for February and was the only profile which contained a mixed layer.

TABLE VI - BILINEAR WITH MIXED LAYER SSP'S (PACIFIC OCEAN)

SSP Number	(30)	RT	ICAPS	CALC
17	17.3	14.2	13.4	8.0

Table VII present the percent error from the averaged ICAPS data and equation (30), ray tracing and CALC.

TABLE VII - BILINEAR WITH MIXED LAYER SSP -
PERCENT ERRORS FROM ICAPS

(30)	RT	CALC
29.1	6.1	40.3

The unusually high percentage error for (30) in this case is a direct repercussion of setting the surface sound speed equal to the sound axis speed. In fact there was about a 12 m/s difference existed between c_2 and c1.

V. CONCLUSIONS

A. MODEL LIMITATIONS

Having an estimate for G based upon a bilinear or bilinear with mixed layer SSP would be sufficient if only those specific SSP's existed. Previous models have had the same limitations of how to segment the SSP [Ref.1:p.78]. A balance must be maintained between practicality and desired level of accuracy. For instance in this research when an isospeed layer was inserted into the bilinear SSP, analysis for θ_{in} quickly became unmanageable. Notice in (30) how the addition of a simple mixed layer to the bilinear SSP added to the complexity of (13). The addition of another layer requires two more parameters to be introduced and will increase the complexity of the gain equation. This becomes quite complicated. Though higher speed computers (to get even faster in the near future) can perform complicated algorithms swiftly and with those algorithms can come even closer to an exact solution to ray tracing, the intention was not to develop an exact solution but a good estimate.

This model, as in previous work [Ref.1:p.78], does not check for the bottom limited sound ray. If the bottom topography interfered with part of the beam the Δr_{CZ} would change and thus G.

As noted earlier, with the sound speeds relatively close (less than about 2 m/s), the model becomes invalid.

The estimates were developed for a source and receiver on the surface. Consequently, the model has no dependency upon source or receiver depth.

The bottom gradient is fixed at 0.017 m/s/m. Little effort would be required to make this change, but the operator would be required to input more data. Additionally, the sound speed at the bottom depth may not be known.

CZGAIN will not run on all MS-DOS machines. High resolution graphics capability is required, but the base equations (10), (13), (26), and (30) can be coded into practically any computer, even a calculator or a Macintosh.

The surface sound speed and sound speed at the sound axis must be equal for (30) to be valid. Minor deviations from this requirement may still be evaluated with (30), but the extent of the deviation allowable is unknown.

B. MODEL PRACTICALITY

In determining practicality, the desirable qualities presented in the introduction will be considered.

1. Data Requirements

The user/operator is required to have an SSP simple enough so that no more than three sound speeds and three depths are needed. If the SSP can be approximated by a bilinear profile, the mixed layer information may be omitted.

2. Output

The output of CZGAIN gives the user a repeat of the inputs in text and graphical formats. Additionally, graphical information on the width of the CZ and range to the inner edge of the CZ are presented.

3. Speed

CZGAIN runs in less time than it takes to input the data. Any of the input data may be quickly modified without having to reenter all data. Earlier models took in excess of nine minutes once the data has been entered [Ref.1:p.80].

4. Program Foundation

Ray tracing theory and not general guidelines or thumbrules were used in the development of the models.

C. COMPARISON TO EARLIER MODELS

The results from the model agreed closely with the values obtained from larger more complicated models (ICAPS). Furthermore, the estimates for G developed were closer in value to ICAPS than earlier models [Ref.1:p.75]. This is particularly interesting since earlier models used a more highly segmented SSP and used an iterative solution rather than an analytical one [Ref.1:p.50]. Also, a tighter requirement was placed on this model for an acceptable difference, 3,dB *vice* 7.3 dB [Ref.1:p.81]. Validation was not performed on the programs developed previously [Ref.1] so no validation comparison was able to be made.

D. GENERAL

The estimates for CZ gain were developed using ray tracing theory. Equation (13) is the estimated solution for G for the bilinear and bilinear with isospeed layer SSP's. Using various combinations of sound speeds and gradients, and thickness of the isospeed layer where appropriate, the solution for G obtained from (13) agreed more closely with the solution for G from ray tracing as well as ICAPS. Equation (30) is the estimated solution for G for the bilinear with mixed layer SSP. This equation was developed under the condition of the surface and sound axis sound speeds being equal. As long as this condition was met, the solution from (30) and solution from ray tracing agreed fairly well (no ICAPS data was available). As long as the SSP's do vary from this requirement, equations (13) and (30) give a reasonable estimates of the convergence zone gain.

LIST OF REFERENCES

1. Badger, R. L., *Convergence Zone Prediction Models with Programs for Use on HP-67 and HP-97 Programmable Calculators*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1979.

2. Urick, R. J., *Principles of Underwater Sound*, 3rd Ed., McGraw-Hill Book Company, 1983.

3. Kinsler, L. E., *et al*, *Fundamentals of Underwater Acoustics*, 3rd ed., John Wiley & Sons, Inc., 1982.

4. Defense Advanced Research Projects Agency, Office of the Secretary of Defense, *Sound Propagation in the Sea*, 1979.

APPENDIX A - SERIES EXPANSIONS

Several expansions are used throughout the development of various equations. The following is a synopsis of those used.

A. TRIGONOMETRIC SERIES EXPANSIONS

1. Cosine θ

$$\cos\theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} - \cdots$$

2. Tangent θ

$$\tan\theta = \theta + \frac{\theta^3}{3} + \frac{2\theta^5}{15} + \frac{17\theta^7}{315} + \frac{62\theta^9}{2835} + \cdots$$

$$\text{where } |\theta| < \frac{\pi}{2}$$

B. BINOMIAL SERIES EXPANSION

$$\left(\frac{1}{1-\theta}\right)^2 = 1 + 2\theta + 3\theta^2 + 4\theta^3 + 5\theta^4 + 6\theta^5 + \cdots$$

$$\text{where } \theta^2 < 1$$

APPENDIX B

```
CC RESWPT2.FOR
CC
CC Program to evaluate the estimation angle to the inner edge of
CC the reswept zone and the CZ gain.  An iteration of the range
CC equation is performed to obtained the true value.  Finally, the
CC percent error of the angle of the ray to the inner edge and CZ gain
CC as compared to the iteration value are calculated.
CC
CC by Al Jenkins
CC
       REAL*8 c_0, c_1, g_0, g_1, R_0, R_1, theta_0, theta_1,
     +           range_e_min, range_i_min, temp_range, G, gamma,
     +           range_outer, theta_e, theta_i,
     +           Gain_iter, Gain_est
      LOGICAL theta_min
      INTEGER i, counter

      OPEN (UNIT=1, ACCESS= 'APPEND', FILE= 'reswpt2a.dat',
     +        STATUS= 'UNKNOWN')
      OPEN (UNIT=2, ACCESS= 'APPEND', FILE= 'reswpt2b.dat',
     +        STATUS= 'UNKNOWN')
      OPEN (UNIT=3, ACCESS= 'APPEND', FILE= 'reswpt2c.dat',
     +        STATUS= 'UNKNOWN')

      WRITE(1,9000)

       theta_min = .FALSE.
       counter = 0

       g_1 = 0.003
       DO 400 m = 1, 10
       WRITE( *,* ) m
       c_1 = 1475
       DO 300 l = 1, 10
       g_0 = .017
       DO 200 k = 1, 10
       c_0 = 1489
      DO 100 j = 1,10

      IF( c_0 .GT. c_1 ) THEN

      G = 1 + g_0 / g_1
      gamma = c_1 / c_0

      theta_min = .FALSE.

      theta_0 = SQRT( (1-gamma**2) / (G**2-1) )
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      theta_e = theta_0
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )

      range_e_min = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +      R_1 * sin(theta_1)

      theta_0 = 0
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )
      range_outer = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +      R_1 * sin(theta_1)
      temp_range = range_outer

     DO 10 i=1,10000
     IF( .NOT. theta_min ) THEN
      theta_0 = theta_0 + 0.000035
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )
      range_i_min  = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +      R_1 * sin(theta_1)
CC Loops set up so as to write only the minimum angle and range
      IF (range_i_min.LT.temp_range) THEN
        temp_range=range_i_min
      ELSE IF (.NOT. theta_min) THEN
        range_i_min = temp_range
        theta_min=.TRUE.
        theta_i = theta_0
      ENDIF
      ENDIF
```

```
 10    CONTINUE

       error = (theta_i - theta_e) / theta_i
       error = 100 * ABS( error)
       WRITE(2,9010) error
       IF( error .GE. 5 )
     +       WRITE(1,9001) c_0, g_0, c_1, g_1, theta_e, theta_i, error
       counter = counter + 1

       Gain_iter = 10 * LOG10( 2* range_i_min /
     +      ( range_outer - range_i_min) )
       Gain_est  = 10 * LOG10( 2* SQRT( G**2 - 1) / (G - SQRT(G**2 -1)))
       error = (Gain_iter - Gain_est) / Gain_iter
       error = 100 * ABS( error)
       WRITE(3,9010) error
       IF( (error*Gain_iter) .GE. 3 )
     +       WRITE(1,9002) Gain_iter, Gain_est, theta_e, theta_i, error
       END IF

       c_0 = c_0 + 6
 100   CONTINUE
       g_0 = g_0 + 0.0229
 200   CONTINUE
       c_1 = c_1 + 5
 300   CONTINUE
       g_1 = g_1 + 0.002
 400   CONTINUE

       WRITE( 1, * ) counter
       CLOSE (1, STATUS= 'KEEP')
       CLOSE (2, STATUS= 'KEEP')
 9000  FORMAT (1X,'c0',8X,'g0',8X,'c1',8X,'g1',8X,'t-e',6X,'t-i',6X,
     +     '% Error')
 9001  FORMAT (1X,F6.1,3X,F5.3,6X,F5.0,4X,F5.3,6X,F5.4,4X,F5.4,4X,F9.6)
 9002  FORMAT (1X,F6.2,3X,F6.2,6X,F5.4,4X,F5.4,4X,F9.6)
 9010  FORMAT (1X,F13.8)
 9100  FORMAT( 1X, F7.3, 4X, F7.3, 3X, F6.1, 3X, F5.3, 6X, F5.0, 4X,
     +     F5.3, 2X, F4.1 )
 9101  FORMAT( 1X, F4.1, 4X, F4.1)
      END
```

APPENDIX C

```
CC  RESWPT3.FOR
CC
CC  Program to evaluate the estimation angle to the inner edge of
CC  the reswept zone and the CZ gain.  An iteration of the range
CC  equation is performed to obtained the true value.  Finally, the
CC  percent error of the angle of the ray to the inner edge and CZ gain
CC  as compared to the iteration value are calculated.  h is the thickness
CC  of the isospeed layer
CC
CC  by Al Jenkins
CC
       REAL*8 c_0, c_1, g_0, g_1, R_0, R_1, theta_0, theta_1,
     +            range_e_min, range_i_min, temp_range, G, gamma,
     +            range_outer, theta_e, theta_i,
     +            Gain_iter, Gain_est, h
      LOGICAL theta_min
      INTEGER i, j, k, l, m, n, counter

      OPEN (UNIT=1, ACCESS= 'APPEND', FILE= 'reswpt3a.dat',
     +        STATUS= 'UNKNOWN')
      OPEN (UNIT=2, ACCESS= 'APPEND', FILE= 'reswpt3b.dat',
     +        STATUS= 'UNKNOWN')
      OPEN (UNIT=3, ACCESS= 'APPEND', FILE= 'reswpt3c.dat',
     +        STATUS= 'UNKNOWN')

      WRITE(1,9000)

       theta_min = .FALSE.
       counter = 0
       h=0

       DO 500 n = 1,10
       g_1 = 0.003
       DO 400 m = 1, 10
       WRITE( *,* ) n,m
       c_1 = 1475
       DO 300 l = 1, 10
       g_0 = .017
      DO 200 k = 1, 10
      c_0 = 1489
      DO 100 j = 1,10

      IF( c_0 .GT. c_1 ) THEN

      G = 1 + g_0 / g_1
      gamma = c_1 / c_0

      theta_min = .FALSE.

      theta_0 = SQRT( (1-gamma**2) / (G**2-1) )
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      theta_e = theta_0
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )

      range_e_min = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +       R_1 * sin(theta_1) + h * cotan( theta_1 )

      theta_0 = 0
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )
      range_outer = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +       R_1 * sin(theta_1) + h * cotan( theta_1 )
      temp_range = range_outer

      DO 10 i=1,10000
      IF( .NOT. theta_min ) THEN
      theta_0 = theta_0 + 0.000035
      theta_1 = acos( c_1 * cos(theta_0) / c_0 )
      R_0     = c_0 / ( g_0 * cos(theta_0) )
      R_1     = c_1 / ( g_1 * cos(theta_1) )
      range_i_min  = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +       R_1 * sin(theta_1) + h * cotan( theta_1 )
CC Loops set up so as to write only the minimum angle and range
      IF (range_i_min.LT.temp_range) THEN
        temp_range=range_i_min
      ELSE IF (.NOT. theta_min) THEN
        range_i_min = temp_range
        theta_min=.TRUE.
        theta_i = theta_0
        ENDIF
        ENDIF
10       CONTINUE

        error = (theta_i - theta_e) / theta_i
        error = 100 * ABS( error)
        WRITE(2,9010) error
        IF( error .GE. 5 )
      +          WRITE(1,9001) c_0, g_0, c_1, g_1, theta_e, theta_i, error
        counter = counter + 1

        Gain_iter = 10 * LOG10( 2* range_i_min /
      +          ( range_outer - range_i_min) )
        Gain_est  = 10 * LOG10( 2* SQRT( G**2 - 1) / (G - SQRT(G**2 -1)))
        error = (Gain_iter - Gain_est) / Gain_iter
        error = 100 * ABS( error)
        WRITE(3,9010) error
        IF( (error*Gain_iter) .GE. 3 ) THEN
           WRITE(1,9001) c_0, g_0, c_1, g_1, h
           WRITE(1,9002) Gain_iter, Gain_est, theta_e, theta_i, error
        ENDIF
        END IF

        c_0 = c_0 + 5.2
100      CONTINUE
        g_0 = g_0 + 0.0229
200      CONTINUE
        c_1 = c_1 + 5
300      CONTINUE
        g_1 = g_1 + 0.002
400      CONTINUE
        h = h + 25
500      CONTINUE

        WRITE( 1, * ) counter
        CLOSE (1, STATUS= 'KEEP')
        CLOSE (2, STATUS= 'KEEP')
9000     FORMAT (1X,'c0',8X,'g0',8X,'c1',8X,'g1',8X,'t-e',6X,'t-i',6X,
      +          '% Error')
9001     FORMAT (1X,F6.1,3X,F5.3,6X,F5.0,4X,F5.3,6X,F5.1,4X,F5.4,4X,F9.6)
9002     FORMAT (1X,F6.2,3X,F6.2,6X,F5.4,4X,F5.4,4X,F9.6)
9010     FORMAT (1X,F13.8)
9100     FORMAT( 1X, F7.3, 4X, F7.3, 3X, F6.1, 3X, F5.3, 6X, F5.0, 4X,
    +    F5.3, 2X, F4.1 )
9101  FORMAT( 1X, F4.1, 4X, F4.1)
    END
```

APPENDIX D

```
CC  RESWPT4.FOR
CC
CC  Program to test the estimated value for theta against the
CC  iterative value of theta for the inner edge of the reswept zone
CC  of a convergence zone.
CC
CC  by Al Jenkins
CC


       REAL*8 c_0, c_1, g_0, g_1, R_0, R_1, theta_0, theta_1,
     +          range_e_min, range_i_min, temp_range, G, gamma,
     +          range_outer, theta_e, theta_i, height, c_2, g_2,
     +          H, eta, R_2
      LOGICAL theta_min
      INTEGER i, j, k, l, m, n, o, p, counter

      OPEN (UNIT=1, ACCESS= 'APPEND', FILE= 'RESWPT4A.DAT',
     +         STATUS= 'UNKNOWN')
      OPEN (UNIT=2, ACCESS= 'APPEND', FILE= 'RESWPT4B.DAT',
     +         STATUS= 'UNKNOWN')
      OPEN (UNIT=3, ACCESS= 'APPEND', FILE= 'RESWPT4C.DAT',
     +         STATUS= 'UNKNOWN')

      WRITE(1,9000)

       theta_min = .FALSE.
       counter = 0


       g_2 = 0.01
       DO 700 p = 1,10

       c_2 = 1475
       DO 600 o = 1,10

       height = 0
       DO 500 n = 1,1
```

```
 g_1 = 0.008
 DO 400 m = 1, 5
 WRITE( *,* ) m, n, o, p

 c_1 = 1475
 DO 300 l = 1, 1

 g_0 = .01
 DO 200 k = 1, 1

 c_0 = 1476
 DO 100 j = 1,10

 IF( c_0 .GT. c_1 .AND. c_0 .GT. c_2) THEN

 G     = 1 + g_0 / g_1
 gamma = c_1 / c_0
 H     = 1 + g_0 / g_2
 eta   = c_2 / c_0

 theta_min = .FALSE.

 theta_0 = H * SQRT( 1 - gamma**2) / SQRT( ( G + g_0/g_2)**2 -
+       H**2 )
 theta_1 = acos( gamma * cos(theta_0) )
 theta_2 = acos( eta * cos(theta_0) )
 theta_e = theta_0
 R_0     = c_0 / ( g_0 * cos(theta_0) )
 R_1     = c_1 / ( g_1 * cos(theta_1) )
 R_2     = c_2 / ( g_2 * cos(theta_2) )
 range_e_min = R_0 * ( sin(theta_1) - sin(theta_0) ) +
+       R_1 * sin(theta_1) + R_2 * ( sin(theta_2) - sin(theta_0) )

 theta_0 = 0
 theta_1 = acos( gamma * cos(theta_0) )
 theta_2 = acos( eta * cos(theta_0) )
 R_0     = c_0 / ( g_0 * cos(theta_0) )
 R_1     = c_1 / ( g_1 * cos(theta_1) )
 R_2     = c_2 / ( g_2 * cos(theta_2) )
 range_outer = R_0 * ( sin(theta_1) - sin(theta_0) ) +
+       R_1 * sin(theta_1) + R_2 * ( sin(theta_2) - sin(theta_0) )
 temp_range = range_outer
```

```
      DO 10 i=1,10000
      IF( .NOT. theta_min ) THEN
       theta_0 = theta_0 + 0.000035
       theta_1 = acos( gamma * cos(theta_0) )
       theta_2 = acos( eta * cos(theta_0) )
       R_0     = c_0 / ( g_0 * cos(theta_0) )
       R_1     = c_1 / ( g_1 * cos(theta_1) )
       R_2     = c_2 / ( g_2 * cos(theta_2) )
       range_i_min  = R_0 * ( sin(theta_1) - sin(theta_0) ) +
     +      R_1 * sin(theta_1) + R_2 * ( sin(theta_2) - sin(theta_0) )
CC Loops set up so as to write only the minimum angle and range
       IF (range_i_min.LT.temp_range) THEN
         temp_range=range_i_min
       ELSE IF (.NOT. theta_min) THEN
         theta_min=.TRUE.
         theta_i = theta_0
       ENDIF
       ENDIF
10      CONTINUE
       IF( .NOT. theta_min ) WRITE( 1,* ) counter
       error = (theta_i - theta_e) / theta_i
       error = 100 * ABS( error)
       WRITE(2,9010) error
       IF( error .GE. 5 )
     +         WRITE(1,9001) c_0, g_0, c_1, g_1, theta_e, height, error
       counter = counter + 1

       Gain_iter = 10*LOG10(2*range_i_min/(range_outer-range_i_min)
       Gain_est = 10*LOG10(2*SQRT((G+g_0/g_2)**2-H**2)/((G+g_0/g_2)-
     +      SQRT((g+g_0/g_2)**2-H**2))
       error = (Gain_iter - Gain_est) / Gain_iter
       error = 100 * ABS( error)
       WRITE(3,9010) error
       IF( (error*Gain_iter) .GE. 3 ) THEN
          WRITE(1,9001) c_0, g_0, c_1, g_1, height
          WRITE(1,9002) Gain_iter, Gain_est, theta_e, theta_i, error
       ENDIF

       ENDIF


       c_0 = c_0 + 7.5
100     CONTINUE
```

```
      g_0 = g_0 + 0.04
200    CONTINUE

      c_1 = c_1 + 5
300    CONTINUE

      g_1 = g_1 + 0.004
400    CONTINUE

      height = height
500    CONTINUE

      c_2 = c_2 + 5
600    CONTINUE

      g_2 = g_2 + 0.005
700    CONTINUE

      WRITE( 1, * ) counter
      CLOSE (1, STATUS= 'KEEP')
      CLOSE (2, STATUS= 'KEEP')
9000   FORMAT (1X,'c0',8X,'g0',8X,'c1',8X,'g1',8X,'t-e',6X,'h',6X,
     +        '% Error')
9001   FORMAT (1X,F6.1,3X,F5.3,6X,F5.0,4X,F5.3,6X,F5.4,4X,F5.0,4X,F13.6)
9002   FORMAT (1X,F6.2,3X,F6.2,6X,F5.4,4X,F5.4,6X,F13.6)
9010   FORMAT (1X,F13.8)
      END
```

APPENDIX E

```
CC  GAIN.FOR
CC
CC  Program for finding the CZ gain
CC
CC  by Al Jenkins
CC

      INCLUDE ’FGRAPH.FI’
      INCLUDE ’FGRAPH.FD’

      LOGICAL fourcolors
      EXTERNAL fourcolors
      REAL*8 c0, c1, c2, h, d0, d1, d3

c     WRITE (*,*) ’ Enter c0 - ’
c     READ (*,*) c0
c     WRITE (*,*) ’ Enter d0 - ’
c     READ (*,*) d0
c     WRITE (*,*) ’ Enter c1 - ’
c     READ (*,*) c1
c     WRITE (*,*) ’ Enter d1 - ’
c     READ (*,*) d1
c     WRITE (*,*) ’ Enter c2 - ’
c     READ (*,*) c2
c     WRITE (*,*) ’ Enter h - ’
c     READ (*,*) h
c     WRITE (*,*) ’ Enter d3 - ’
c     READ (*,*) d3

      c0 = 1500
      c1 = 1475
      c2 = 1490
      h  = 250
      d0 = 100
      d1 = 500
      d3 = 3000
      IF ( fourcolors() ) THEN
         CALL plot_all(c0, c1, c2, h, d0, d1, d3)
      ELSE
         WRITE (*,*) ' This program requires...'
      END IF
      END
CC
CC
CC  FOURCOLORS -
CC  p.180 Advanced Topics Guide
CC  Microsoft Corporation
CC
      LOGICAL FUNCTION fourcolors()

      INCLUDE 'FGRAPH.FD'

      INTEGER*2 dummy
      RECORD /videoconfig/ screen
      COMMON screen

C
C   Set to max number of available colors.
C

      CALL getvideoconfig( screen )
      SELECT CASE( screen.adapter )
         CASE( $CGA, $OCGA )
           dummy = setvideomode( $MRES4COLOR )
         CASE( $EGA, $OEGA )
           dummy = setvideomode( $ERESCOLOR )
         CASE( $VGA, $OVGA )
           dummy = setvideomode( $VRES16COLOR )
         CASE DEFAULT
           dummy = 0
      END SELECT

      CALL getvideoconfig( screen )
      fourcolors = .TRUE.
      IF( dummy .EQ. 0 ) fourcolors = .FALSE.
      END
CC
CC
```

```
CC  PLOT_ALL   Displays the graphs
CC
CC
      SUBROUTINE plot_all(c0, c1, c2, h, d0, d1, d3)

      INCLUDE 'FGRAPH.FD'

      REAL*8 c0, c1, c2, h, d0, d1, d3
      INTEGER*2 dummy, halfx, halfy, xwidth, yheight
      INTEGER*2 cols, rows
      RECORD /videoconfig/ screen
      COMMON screen
      CHARACTER*1 string, getch
      RECORD /rccoord/ curpos

10    CONTINUE

C
C   Find screen dimensions
C
      CALL clearscreen ( $GCLEARSCREEN )
      xwidth     = screen.numxpixels
      yheight = screen.numypixels
      cols  = screen.numtextcols
      rows       = screen.numtextrows
      halfx      = xwidth/2 * 0.8
      halfy      = (yheight/rows)*(rows/2)
C
C  Sound Speed window
C
      CALL setviewport( 0, 0, halfx , halfy )
      CALL settextwindow( 1, 1, rows / 2, cols / 2  )
      dummy = setwindow( .TRUE., -2.0, -1.2, 2.0, 1.75 )
      CALL sound_speed(c0, c1, c2, h, d0, d1, d3)
      dummy = rectangle ($GBORDER, 0, 0, halfx - 4, halfy - 2 )
C
C  Range vs. Theta window
C
      CALL setviewport( halfx - 2, 0, xwidth-1, yheight - 1 )
      CALL settextwindow( 1, (cols/2-5) , rows, cols )
      dummy = setwindow( .TRUE., -2.0, -2.0, 2.0, 2.0 )
      CALL range_plot(c0, c1, c2, h, d0, d1)
      dummy = rectangle_w( $GBORDER, -2.0, -2.0, 2.0, 2.0 )
```

```
C
C  Graph Description Window
C
      CALL setviewport( 0, halfy+1 , halfx-1 , yheight-1)
      CALL settextwindow( rows / 2 + 1, 0, rows, cols / 2 )
      dummy = setwindow( .TRUE., -2.0, -1.2, 2.0, 1.75 )
      CALL graph_description( c0, d0, c1, d1, d3, h, c2 )
      dummy = rectangle ($GBORDER, 0, 0, halfx - 4, halfy - 2 )

C
C   Check for increasing, adding, rasing sound speed at surface
C
      CALL settextposition( 0, 0, curpos )
      string = getch()
      SELECT CASE( string )
      CASE( '1' )
         GOTO 20
      CASE( '!', 'A', 'a' )
         GOTO 25
      CASE( '2' )
         GOTO 30
      CASE( '@', 'S', 's' )
         GOTO 35
      CASE( '3' )
         GOTO 40
      CASE( '#', 'D', 'd' )
         GOTO 45
      CASE( '4' )
         GOTO 50
      CASE( '$', 'F', 'f' )
         GOTO 55
      CASE( '5' )
         GOTO 60
      CASE( '%', 'G', 'g' )
         GOTO 65
      CASE( '6' )
         GOTO 70
      CASE( '^', 'H', 'h' )
         GOTO 75
      CASE( 'E', 'e', 'Q', 'q')
         GOTO 100
      CASE DEFAULT
         GOTO 10
      END  SELECT
      GOTO 10

20    c2 = c2 - 1
      IF (c2.LE.c1)  c2 = c2 + 1
      IF (c2.LE.0 )  c2 = c2 + 1
      GOTO 10
25    c2 = c2 + 1
      IF (c2.GE.c0)  c2 = c2 - 1
      GOTO 10
30    c0 = c0 - 1
      IF (c0.LE.c2)  c0 = c0 + 1
      IF (c0.LE.c1)  c0 = c0 + 1
      IF (c0.LE.0 )  c0 = c0 + 1
      GOTO 10
35    c0 = c0 + 1
      GOTO 10
40    d0 = d0 - 5
      IF (d0.LE.0) d0 = d0 + 5
      GOTO 10
45    d0 = d0 + 5
      IF (d0.GE.d1) d0 = d0 - 5
      GOTO 10
50    c1 = c1 - 1
      IF (c1.LE.0 ) c1 = c1 + 1
      GOTO 10
55    c1 = c1 + 1
      IF (c1.GE.c2) c1 = c1 - 1
      IF (c1.GE.c0) c1 = c1 - 1
      GOTO 10
60    d1 = d1 - 5
      IF( d1 .LE. d0 ) d1 = d1 + 5
      GOTO 10
65    d1 = d1 + 5
      IF (d1.GE.(d3 - h)) d1 = d1 - 5
      GOTO 10
70    h = h - 5
      IF( h .LT. 0 ) h = h + 5
      GOTO 10
75    h = h + 5
      IF( h .GE. (d3 - d1 - d0) ) h = h - 5
      GOTO 10
```

```
100  CONTINUE
        dummy = setvideomode ( $DEFAULTMODE )
     END


CC
CC
CC SOUND_SPEED Subroutine to plot the sound speed profile
CC
CC
     SUBROUTINE sound_speed(c0, c1, c2, h, d0, d1, d3)

     INCLUDE 'FGRAPH.FD'

     REAL*8 c0, c1, c2, h, d0, d1, d3
     INTEGER*2 dummy
     DOUBLE PRECISION x, y, scale_x
     RECORD /videoconfig/ screen
     RECORD /wxycoord/ wxy
     RECORD /rccoord/ curpos
     COMMON screen
     CHARACTER*8 string

     dummy = settextcolor( 15 )
     CALL settextposition( 8, 2, curpos )
     CALL outtext( 'Depth' )
     CALL settextposition( 2, 10, curpos )
     CALL outtext( 'Sound Speed' )
     CALL settextposition( 3, 12, curpos )
     dummy = settextcolor( 14 )
     WRITE (string, '(I4)')INT( c2 )
     CALL outtext( ' ' // string // ' ')
     CALL settextposition( 3, 19, curpos )
     dummy = settextcolor( 12 )
     WRITE (string, '(I4)')INT( c0 )
     CALL outtext( ' ' // string // ' ')

C
C  Draw border
C
     dummy = setcolor( 1 )
     dummy = rectangle_w( $GBORDER, -1.00, -1.00, 1.00, 1.00 )
     dummy = rectangle_w( $GBORDER, -1.02, -1.02, 1.02, 1.02 )
```

```
C
C  Plot points
C
          scale_x = - 50.0
          dummy = setcolor( 14 )
          CALL    moveto_w( 0, 1, wxy )
          x = scale_x * ( 1 - c0 / c2 )
          y = 1 - 2 * d0 / d3
          dummy = lineto_w( x, y )
          CALL    moveto_w( x, y, wxy )
          x = scale_x * ( 1 - c1 / c2 )
          y = 1 - 2 * d1 / d3
          dummy = lineto_w( x, y )
          CALL    moveto_w( x, y, wxy )
          y = 1 - 2 * ( d1 + h ) / d3
          dummy = lineto_w( x, y )
          CALL    moveto_w( x, y, wxy )
          x = scale_x *(1-(c1+0.017*(d3-d1-h))/c2)
          dummy = lineto_w( x, -1 )
          dummy = setcolor( 3 )
C
C  Plot points for bilinear profile
C
          dummy = setcolor( 12 )
          x = scale_x * ( 1 - c0 / c2 )
          CALL    moveto_w( x, 1, wxy )
          x = scale_x * ( 1 - c1 / c0 )
          y = 1 - 2 * d1 / d3
          dummy = lineto_w( x, y )
          CALL    moveto_w( x, y, wxy )
          x = scale_x *(1-(c1+0.017*(d3-d1))/c0)
          dummy = lineto_w( x, -1 )
      dummy = setcolor( 3 )
      END
CC
CC
CC RANGE_PLOT Subroutine for plotting range versus theta
CC
CC
      SUBROUTINE range_plot(c0, c1, c2, h, d0, d1)

      INCLUDE 'FGRAPH.FD'
      INTEGER*2 dummy, i
      DOUBLE PRECISION x, y, theta_2, range_scale
      DOUBLE PRECISION theta, theta_graph, range, range_graph
      REAL*8 unit_dimension, range_12, theta_12, gamma, G
      REAL*8 range_14, theta_14
      REAL*8 c0, c1, c2, d0, d1, h, scale_r, scale_t
      RECORD /videoconfig/ screen
      RECORD /wxycoord/ wxy
      RECORD /rccoord/ curpos
      COMMON screen
      CHARACTER*8 string
C
C   Label grid
C
         dummy = settextcolor( 13 )
      CALL settextposition( 14, 0, curpos )
         CALL outtext( 'Range' )
      CALL settextposition( 15, 8, curpos )
         CALL outtext( ' 0' )
      CALL settextposition( 10, 8, curpos )
         CALL outtext( ' 2' )
      CALL settextposition(  4, 8, curpos )
         CALL outtext( ' 4' )
      CALL settextposition( 21, 8, curpos )
         CALL outtext( '-2' )
      CALL settextposition( 27, 8, curpos )
         CALL outtext( '-4' )

      CALL settextposition( 35, 11, curpos )
      CALL outtext( '0      2      4      6      8 Theta' )

C
C  Draw border
C
      dummy = setcolor( 1 )
      dummy = rectangle_w( $GBORDER, -1.00, -1.80, 1.90, 1.90 )
      dummy = rectangle_w( $GBORDER, -1.02, -1.82, 1.92, 1.92 )
C
C  Draw Grid
C
      x = -0.71
      y = -1.48
      DO i = 1, 9
```

```
            dummy = setcolor( 8 )
            CALL    moveto_w( x, -1.8, wxy )
            dummy = lineto_w( x, 1.9 )
            CALL    moveto_w( -1.0, y, wxy )
            dummy = lineto_w( 1.9, y )
            y       = y + 0.37
            x       = x + 0.29
         END DO
C
C  Plot curve
C
         g0 = ABS( (c0 - c1) / (d1 - d0) )
         g1 = ABS( 0.017 )
         g2 = ABS( (c1 - c2) / d0 )

C  Set scaling factor for vertical axis, based on bilinear profile
         gamma = c1 / c0
         G     = 1 + g0 / g1
         theta = SQRT( ( 1 - gamma**2 ) / ( G**2 - 1 ) )
         theta_1     = acos (c1 * cos(theta) / c0)
         theta_2     = acos (c2 * cos(theta) / c0)

c        range=(c0/(g0*cos(theta)))*((1+g0/g1)*DSQRT(1-(c1/c0)**2 *
c     +        cos(theta)**2) - sin(theta)) +
c     +        h*cotan(acos((c1/c0)*cos(theta)))
         range = c2/(g2*cos(theta_2)) * (sin(theta_2) - sin(theta)) +
     +           c0/(g0*cos(theta)) * (sin(theta_1) - sin(theta)) +
     +           h * cotan(theta_1) +
     +           c1/(g1*cos(theta_1)) * sin(theta_1)
         range_scale = 0.045 * range

         dummy = setcolor( 14 )
         scale_r     = 1.0
C  Set horizontal scale so that 1.5 degrees equals 0.1 across the
C  total width of 2.9
         scale_t     = 11.07718404
         theta_graph = -1.0
         range_graph = 0
         theta        = 0
         theta_1     = acos (c1 * cos(theta) / c0)
         theta_2     = acos (c2 * cos(theta) / c0)

         g0 = ABS( (c0 - c1) / (d1 - d0) )
```

```
      g1 = ABS( 0.017 )
      g2 = ABS( (c0 - c2) / d0 )

      range = c2/(g2*cos(theta_2)) * (sin(theta_2) - sin(theta)) +
     +        c0/(g0*cos(theta)) * (sin(theta_1) - sin(theta)) +
     +        h * cotan(theta_1) +
     +        c1/(g1*cos(theta_1)) * sin(theta_1)

C   Vertical dimension for each graph square in meters
      unit_dimension = range_scale / scale_r

C  Dimensions for the graph for the full profile in meters and radians
      range_14      = 2 * range
      theta_14      = theta_2

      DO i = 1, 100
        CALL moveto_w( theta_graph, range_graph, wxy )
        theta    = theta+.0026179939
        theta_1 = acos (c1 * cos(theta) / c0)
        theta_2 = acos (c2*cos(theta)/c0)
        range_graph =
     +        c2/(g2*cos(theta_2)) * (sin(theta_2) - sin(theta)) +
     +        c0/(g0*cos(theta)) * (sin(theta_1) - sin(theta)) +
     +        h * cotan(theta_1) +
     +        c1/(g1*cos(theta_1)) * sin(theta_1)
        range_graph=(range_graph - range)/range_scale
C       Scaling factor for the graph
        range_graph= scale_r * range_graph
        theta_graph= scale_t * theta - 1
        dummy = lineto_w( theta_graph, range_graph )
      END DO
C
C  Plot the non-mixed layer curve
C
      dummy        = setcolor( 12 )
      theta_graph = -1.0
      theta           = 0.0
      range_graph = 0.0
      g0          = ABS( (c0 - c1) / d1 )
      theta_1      = acos(c1*cos(theta)/c0)

        range = c0/(g0*cos(theta)) *
     +        (sin(theta_1) - sin(theta)) +
     +        c0/(g1*cos(theta)) * sin(theta_1 )

C  Dimensions for the graph for the bilinear profile in meters and radians
      range_12      = 2 * range
      theta_12      = 0

      DO i = 1, 100
        CALL moveto_w( theta_graph, range_graph, wxy )
        theta=theta+.0026179939
        theta_1 = acos(c1*cos(theta)/c0)
        range_graph = c0/(g0*cos(theta)) *
     +        (sin(theta_1) - sin(theta)) +
     +        c1/(g1*cos(theta_1)) * sin(theta_1 )
        range_graph = (range_graph - range) / range_scale
C       Scaling factor for graph
        range_graph = scale_r * range_graph
        theta_graph = scale_t * theta - 1
        dummy = lineto_w( theta_graph, range_graph )
      END DO
C
C  Put a plus for the estimated minimum range and its angle
C
      dummy = setcolor( 15 )
      gamma = c1 / c0
      G     = 1 + g0 / g1
      theta = SQRT( ( 1 - gamma**2 ) / ( G**2 - 1 ) )
      range=(c0/(g0*cos(theta)))*((1+g0/g1)*DSQRT(1-(c1/c0)**2 *
     +        cos(theta)**2) - sin(theta))
      range_graph =  ( range / range_12 )  - 1
c     WRITE(*,*) range_12, range
      range_graph = scale_r * range_graph
      theta_graph = scale_t*theta - 1
c     CALL moveto_w( theta_graph-0.05, range_graph, wxy )
c     dummy = lineto_w( theta_graph+0.05, range_graph )
c     CALL moveto_w( theta_graph, range_graph-0.05, wxy )
c     dummy = lineto_w( theta_graph, range_graph+0.05 )
C
C  Reference information for the graph
C
          dummy = settextcolor( 13 )
          CALL settextposition( 2, 14, curpos )
          WRITE (string, '(F4.1)') unit_dimension / 1852
          CALL outtext( 'Unit dimensions:           ')
          CALL settextposition( 3, 13, curpos )
          CALL outtext( ' 1.5 deg by'  // string // 'nmi        ')
          dummy = settextcolor( 12 )
          CALL settextposition( 4, 12, curpos )
          WRITE (string, '(F5.1)') range_12 / 1852
          CALL outtext( 'Range (0) = '// string// 'nmi     ')
          CALL settextposition( 5, 12, curpos )
          WRITE( string, '(F4.2)') theta_12 * 57.29578
          CALL outtext( 'Theta (0) =  '// string// ' ' )
          CALL settextposition( 5, 32, curpos )
          CALL outtext( 'degrees')
          dummy = settextcolor( 14 )
          CALL settextposition( 6, 12, curpos )
          WRITE (string, '(F5.1)') range_14 / 1852
          CALL outtext( 'Range (0) = '// string// 'nmi     ')
          CALL settextposition( 7, 12, curpos )
          WRITE( string, '(F5.2)') theta_14 * 57.29578
          CALL outtext( 'Theta (0) = '// string// ' ' )
          CALL settextposition( 7, 32, curpos )
          CALL outtext('degrees')
          dummy = settextcolor( 15 )
          CALL settextposition( 8, 12, curpos )
          WRITE (string, '(F5.1)') range / 926
          CALL outtext( 'Range (+) = '// string// 'nmi        ')
          CALL settextposition( 9, 12, curpos )
          WRITE( string, '(F5.2)') theta * 57.29578
          CALL outtext( 'Theta (+) = ' // string// ' ' )
          CALL settextposition( 9, 32, curpos )
          CALL outtext('degrees')
      dummy = setcolor( 3 )
      END
CC
CC
CC  Subroutine for the graph description
CC
CC
      SUBROUTINE graph_description( c0, d0, c1, d1, d3, h, c2 )

      INCLUDE 'FGRAPH.FD'

      REAL*8 c0, d0, c1, d1, c3, d3, h, g1, g2, g0, gain_bilinear,
     +        gain_mxdlyr, theta, R0, R1, R2, max_range, CZ_width,theta_1,
     +        theta_2, min_range, temp, c2
```

```
      CHARACTER*8 string
      RECORD /videoconfig/ screen
      RECORD /rccoord/ curpos
      COMMON screen

C
C  Information on the sound speed vs. depth graph
C
         dummy = settextcolor(15)
         CALL settextposition( 2, 2, curpos )
         CALL outtext( ' Depth,m    Sound Speed, m/s')
         dummy = settextcolor(14)
         CALL settextposition( 3, 2, curpos )
         WRITE (string, '(I5)')INT( d0 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 3, 15, curpos )
         WRITE (string, '(I5)')INT( c0 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 4, 2, curpos )
         WRITE (string, '(I5)')INT( d1 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 4, 15, curpos )
         WRITE (string, '(I5)')INT( c1 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 5, 2, curpos )
         WRITE (string, '(I5)')INT( d1 + h )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 5, 15, curpos )
         WRITE (string, '(I5)')INT( c1 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 6, 2, curpos )
         WRITE (string, '(I5)')INT( d3 )
         CALL outtext( ' ' // string // ' ')
         CALL settextposition( 6, 15, curpos )
         c3 = c1+0.017*(d3-d1-h)
         WRITE (string, '(I5)')INT( c3 )
         CALL outtext( ' ' // string // ' ')
C
C  Gain information, bilinear then mixed layer
C
      temp = temp
      gain_mxdlyr = 0
      g0   = ABS( (c0 - c1) / (d1 - d0) )
```

```
 g1   = ABS( 0.017 )
 g2   = ABS( (c1 - c2) / d0 )


 gain_bilinear = 10 * LOG10( (2 * SQRT( (1+g0/g1)**2-1 )) /
+        ( (1+g0/g1) - SQRT( (1+g0/g1)**2-1 )))

 dummy = settextcolor(12)
 CALL settextposition( 8, 2, curpos )
 CALL outtext( ' CZ Gain =')
 CALL settextposition( 8, 12, curpos )
 WRITE (string, '(F4.1)') gain_bilinear
 CALL outtext( ' ' // string //' ')
 CALL settextposition( 8, 18, curpos )
 CALL outtext( 'db')

 gain_mxdlyr = 10 * LOG10( (2 * SQRT( (1+g0/g1+g0/g2)**2-1 )) /
+        ( (1+g0/g1+g0/g2) - SQRT( (1+g0/g1+g0/g2)**2-1 )))
 dummy = settextcolor(14)
 CALL settextposition( 10, 2, curpos )
 CALL outtext( ' CZ Gain =')
 CALL settextposition( 10, 12, curpos )
 WRITE (string, '(F4.1)') gain_bilinear
 CALL outtext( ' ' // string //' ')
 CALL settextposition( 10, 18, curpos )
 CALL outtext( 'db')

 END
```

APPENDIX F

USER'S MANUAL FOR CZ GAIN

CZGAIN requires an MS-DOS based system with high resolution graphics capability (VGA or better). The model was developed using ray tracing rather than normal mode theory.

The user inputs the following data:

- Surface sound speed, m/s
- Depth of mixed layer, m
- Sound speed at bottom of mixed layer, m/s
- Minimum sound speed below the mixed layer, m/s
- Depth of that sound speed, m/s
- Bottom depth, m

While the bottom sound speed gradient is fixed at 0.017 /s, the bottom depth is used when drawing SSP's.

The surface sound speed and minimum sound speed will always be less than the sound speed at the bottom of the mixed layer.

The depth of the mixed layer will be no less than 1 meter.

The thickness of the isospeed layer will be zero or greater.

When developing the keys to use for the data manipulation, efforts were made to tie each data point to a specific mnemonic (e.g. Thickness of isospeed layer changed by t). No standard convention could be established. The numbers work from left to right on sound speed and depth pairs. Where no depth or sound speed was used (e.g. surface depth is always zero) that key was used for the next parameter. The numbers always decrease the respective value and using the SHIFT key with the number always increases the respective value. The letters were added at the programmers personal convenience. No enter key is required, the program automatically captures the keystroke and recomputes the output. The following is a list of keys which will manipulate the data.

- 1 - Decreases surface sound speed
- !, A, a - Increases surface sound speed
- 2 - Decreases sound speed at bottom of mixed layer
- @, S, s - Increases sound speed at bottom of mixed layer
- 3 - Decreases depth of mixed layer
- #, D, d - Increases depth of mixed layer
- 4 - Decreases minimum sound speed
- $, F, f - Increases minimum sound speed
- 5 - Decreases depth of minimum sound speed
- %, G, g - Increases depth of minimum sound speed
- 6 - Deceases thickness of isospeed layer
- ^, H, h - Increases thickness of isospeed layer
- 7 - Decreases bottom depth
- &, J, j - Increases bottom depth
- Q, q, E, e - Exits the program

If the program is modified, the Key Entry Functions (if using Microsoft Fortran) patch will need to be installed into the proper libraries before compiling.

BIBLIOGRAPHY

Beckes, M.E., Burhans, N.P., and Gump, R.E., *Passive Environmental ASW Prediction System (PEAPS)*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1975.

Coppens, A.B, Dahl, H.A., and Sanders, J.V., *An Introduction to the Sonar Equations with Applications (Revised)*, Naval Postgraduate School, February 1979.

Madden, L., *Pocket-Proploss Programs and Models for the Computation of Acoustic Transmission Loss on Hand-Held Calculators*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1977.

Microsoft Corporation, *Microsoft FORTRAN, Advanced Topics*, 1989.

Microsoft Corporation, *Microsoft FORTRAN, Reference*, 1989.

Naval Ocean Systems Center, Technical Report 997, *Comparison of Experimental Detailed Convergence-Zone Structure with Results of Uniform Asymptotic Ray Theory*, by M.A. Pedersen, D.F. Gordon, and R.F. Hosmer, 15 December 1985.

Officer, C., *Introduction to the Theory of Sound Transmission*, McGraw-Hill Book Company, INC., 1958.

Podeszwa, E.M., *A Slide Rule for Predicting Convergence Zones*, U.S. Navy Underwater Sound Laboratory, USL Report No. 877, 22 April 1969.

Shudde, R.H., *Naval Applications: Ten Algorithms for the Hewlett-Packard HP-67 and HP-97 Calculators*, Naval Postgraduate School, Report No. NPS55-79-04, February 1979.

United States Navy Underwater Sound Laboratory, Fort Trumbull, New London, Connecticut, USL Report Number 877, *A Slide Rule for Predicting Convergence Zones*, by E.M. Podeszwa, 22 April 1969.

Urick, R.J., *Sound Propagation in the Sea*, Defense Advanced Research Projects Agency, 1979.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 52 2
Naval Postgraduate School
Monterey, California 93943-5002

3. Associate Professor A.B. Coppens 2
Code PH/Cz
Department of Physics
Naval Postgraduate School
Monterey, California 93943

4. Associate Professor J.V. Sanders 1
Code 33A
Department of Physics
Naval Postgraduate School
Monterey, California 93943

6. Mr. D. Stewart 1
RW-52
Patuxent River NAS, Maryland 20670

5. Mr. A.K. Jenkins 2
RW-52L
Patuxent River NAS, Maryland 20670